



## Θ.Ε. ΠΛΣ50 (2016-17) – ΓΡΑΠΤΗ ΕΡΓΑΣΙΑ Ε2

Ημερομηνία ανάρτησης	14.11.2016
Ημερομηνία αποστολής	<b>Βάσει χρονοδιαγράμματος: 18.12.2016</b> <b>Δεκτή μέχρι την Τετάρτη 21.12.2016, ώρα 23:59</b> (προσοχή, το σύστημα υποβολής θα κλείσει αυτόματα μόλις παρέλθει η παραπάνω χρονική στιγμή, σύμφωνα με το ρολόι του συστήματος, που μπορεί να διαφέρει ελαφρώς από το δικό σας)
Ανακοίνωση ενδεικτικής επίλυσης	23.12.2015

### Θεματολογία-στόχος

Στην εργασία αυτή θα ασχοληθείτε με προχωρημένα θέματα της γλώσσας προγραμματισμού C, ειδικότερα με πέρασμα παραμέτρων σε συναρτήσεις με χρήση αναφοράς, με δείκτες, δείκτες σε συμβολοσειρές, με αναδρομικές συναρτήσεις, με πολυδιάστατους πίνακες, αναζήτηση, ταξινόμηση, απλά συνδεδεμένες λίστες, δημιουργία ψευδοτυχαίων αριθμών καθώς και με σύνθετες δομές δεδομένων της γλώσσας προγραμματισμού C.

### Παρατηρήσεις

Περιμένουμε όλες οι εργασίες να ανεβούν στο study.eap.gr, όπως σας έχει υποδείξει ο σύμβουλος καθηγητής του τμήματός σας. Ο πηγαίος κώδικας C (ένα πρόγραμμα C για κάθε θέμα ή υποερώτημα) που θα συνοδεύει την εργασία θα πρέπει να βρίσκεται σε ξεχωριστά αρχεία (\*.c) (ένα αρχείο \*.c για κάθε θέμα ή υποερώτημα). Μην συμπεριλάβετε εκτελέσιμα (.exe) ή αρχεία object code (.o / .obj). Συνιστάται να συμπεριλάβετε επεξηγήσεις για τον τρόπο που λύσατε την εργασία σε συνοδευτικό αρχείο κειμένου μορφής \*.doc ή \*.odt (αρχεία \*.pdf γίνονται δεκτά μόνο όταν συνοδεύονται από το αντίστοιχο doc/odt). Στο study.eap.gr, σε κάθε περίπτωση, ανεβάζετε **ένα μόνο συμπιεσμένο αρχείο** (\*.zip ή \*.rar) που θα περιέχει όλα τα επιμέρους αρχεία.

### Εισαγωγή

Ο πηγαίος κώδικας των προγραμμάτων σας θα πρέπει να είναι επαρκώς σχολιασμένος και επιπλέον ANSI-συμβατός (δεν γίνονται δεκτές απαντήσεις που κάνουν χρήση συναρτήσεων από επεκτάσεις της C).

Δε χρειάζεται να κάνετε τίποτε περισσότερο από όσα σας ζητούνται. Συνιστάται, πριν ασχοληθείτε με την εργασία, να μελετήσετε τις λύσεις εργασιών προηγούμενων ετών. Στο τέλος της Εργασίας περιλαμβάνεται **ενδεικτικός** πίνακας με τα βασικά γνωστικά αντικείμενα της γλώσσας προγραμματισμού C που πρέπει να έχετε διαβάσει για να μπορείτε να αντιμετωπίσετε κάθε θέμα. Τονίζεται ότι ο πίνακας έχει σκοπό να σας βοηθήσει και όχι να σας υποχρεώσει στον ακριβή τρόπο υλοποίησης κάθε θέματος.

Τα μηνύματα που τυπώνουν τα προγράμματά σας στην οθόνη συνιστάται να είναι γραμμένα με λατινικούς χαρακτήρες (greeklish ή αγγλικά), προς αποφυγή προβλημάτων που σχετίζονται με την κωδικοποίηση των ελληνικών στο λειτουργικό σύστημα του υπολογιστή σας.

### Θέμα 1: Νέο σύστημα καταγραφής κρατημένων θέσεων

Στο 4<sup>ο</sup> θέμα της Γραπτής Εργασίας 1 ακ. έτους 2016-17 ζητούνταν η δημιουργία συστήματος καταγραφής κρατημένων θέσεων λεωφορείου. Στο θέμα αυτό σας ζητείται να δημιουργήσετε μια νέα έκδοση του συστήματος, που θα βασίζεται σε επιπλέον χαρακτηριστικά της C και πιο σύνθετες δομές δεδομένων.

Ειδικότερα, η νέα έκδοση του συστήματος θα πρέπει να έχει τα ακόλουθα χαρακτηριστικά:

Η αναπαράσταση των θέσεων του λεωφορείου να γίνεται με έναν μονοδιάστατο πίνακα 53 θέσεων (μέγιστο πλήθος θέσεων λεωφορείου), όπου ένα στοιχείο του πίνακα αντιστοιχεί σε μια θέση του λεωφορείου. Ο τύπος ενός στοιχείου του πίνακα να είναι ένα struct με πεδία α) το ονοματεπώνυμο (που περιλαμβάνει το επώνυμο, ένα κενό και το όνομα) του επιβάτη, με τύπο πίνακα 40 χαρακτήρων

(συμβολοσειρά 40 θέσεων), με την κενή συμβολοσειρά (μόνο ο χαρακτήρας '\0') να εκφράζει μια ελεύθερη θέση, β) το τηλέφωνο του επιβάτη, με τύπο πίνακα unsigned short 10 θέσεων και γ) τον αριθμό θέσης του επιβάτη, με τύπο unsigned int. Ίσως η αποθήκευση του αριθμού θέσης του επιβάτη να μοιάζει περιττή, καθώς η θέση του πίνακα εκφράζει και τον αριθμό θέσης του επιβάτη, όμως θα σας φανεί χρήσιμη στο επόμενο θέμα.

Κατά την έναρξη του προγράμματος να γίνεται ανάγνωση από αρχείο με όνομα bus.txt της πινακίδας κυκλοφορίας (που αποτελείται από 7 χαρακτήρες, 3 γράμματα και 4 αριθμητικά ψηφία) και του πλήθους των θέσεων του λεωφορείου και τα στοιχεία αυτά τυπώνονται στην οθόνη. Μετά το πλήθος θέσεων, το αρχείο bus.txt περιέχει και τις τυχόν κρατημένες θέσεις του λεωφορείου (μια ξεχωριστή γραμμή για κάθε κρατημένη θέση, με το επώνυμο, ένα κενό, το όνομα, ένα κενό, τον αριθμό θέσης, ένα κενό και το τηλέφωνο του επιβάτη). Σας δίνεται δείγμα του αρχείου bus.txt με μερικές κρατημένες θέσεις. Θεωρήστε ότι η πινακίδα κυκλοφορίας, ο αριθμός των θέσεων και τα στοιχεία των κρατημένων θέσεων στο αρχείο bus.txt είναι έγκυρα, οπότε δεν χρειάζεται το πρόγραμμά σας να κάνει σχετικούς ελέγχους. Ο πίνακας θέσεων πρέπει να αρχικοποιηθεί κατάλληλα ώστε όλες οι θέσεις να είναι κενές.

Κατόπιν, να εμφανίζεται αριθμημένη λίστα επιλογών για

1. εμφάνιση του συνολικού πλήθους κενών θέσεων και ταξινομημένης λίστας των αριθμών τους,
2. κράτηση θέσης με συγκεκριμένο αριθμό για στοιχεία επιβάτη που θα παρέχει ο χρήστης (πρέπει να προηγείται έλεγχος ότι ο αριθμός αυτός είναι εντός ορίων και ότι η θέση δεν είναι ήδη κρατημένη),
3. αναζήτηση για το αν είναι κρατημένη θέση με συγκεκριμένο ονοματεπώνυμο ή τηλέφωνο επιβάτη, ανάλογα με επιπλέον επιλογή του χρήστη,
4. ακύρωση κράτησης θέσης με συγκεκριμένο αριθμό,
5. εμφάνιση λίστας (ονοματεπώνυμο και τηλέφωνο επιβάτη, αριθμός θέσης) κρατημένων θέσεων ταξινομημένης κατά αριθμό θέσης,
0. τερματισμό του προγράμματος.

Κατά την έξοδο από το πρόγραμμά σας, θα πρέπει να αποθηκεύονται εκ νέου η πινακίδα κυκλοφορίας του λεωφορείου, ο αριθμός θέσεων του λεωφορείου και οι κρατημένες θέσεις του ταξινομημένες κατά αριθμό θέσης στο αρχείο bus.txt, με την ίδια μορφοποίηση (ώστε να είναι δυνατή η ανάγνωση του αρχείου σε επόμενη εκτέλεση του προγράμματός σας).

Οι διάφορες λειτουργίες που αντιστοιχούν στη λίστα επιλογών του προγράμματος, καθώς και η ανάγνωση από το / αποθήκευση στο αρχείο bus.txt να διεκπεραιώνονται από διαφορετικές συναρτήσεις. Στην επιλογή 2 και την επιλογή 4 της λίστας επιλογών του προγράμματος να ζητείται από το κυρίως πρόγραμμα ο αριθμός θέσης και με χρήση αυτού του αριθμού να προσδιορίζεται και να περνιέται με αναφορά το struct της θέσης προς κράτηση ή ακύρωση κράτησης στη συνάρτηση κράτησης (επιλογή 2) ή ακύρωσης κράτησης (επιλογή 4), αντίστοιχα.

## Θέμα 2: Επέκταση συστήματος καταγραφής κρατημένων θέσεων

Στο θέμα αυτό σας ζητείται να τροποποιήσετε το σύστημα του προηγούμενου θέματος, βασιζόμενοι σε επιπλέον χαρακτηριστικά της C και συνδεδεμένες δομές δεδομένων.

Φροντίστε ώστε η αναπαράσταση των θέσεων του λεωφορείου να γίνεται με έναν μονοδιάστατο δυναμικό πίνακα, με στοιχεία ίδιου τύπου με το προηγούμενο θέμα. Με βάση τον αριθμό θέσεων του λεωφορείου που διαβάσατε από το αρχείο bus.txt, να εκχωρήσετε δυναμικά μνήμη για τον πίνακα αυτόν, ο οποίος και πάλι πρέπει να αρχικοποιηθεί κατάλληλα ώστε όλες οι θέσεις να είναι κενές.

Επεκτείνετε την επιλογή 5 της λίστας επιλογών ως εξής:

5. εμφάνιση λίστας (ονοματεπώνυμο και τηλέφωνο επιβάτη, αριθμός θέσης) κρατημένων θέσεων ταξινομημένης κατά αριθμό θέσης, ή κατά ονοματεπώνυμο (πρώτα κατά επώνυμο και κατόπιν κατά όνομα), ανάλογα με επιπλέον επιλογή του χρήστη.

Αν ο χρήστης επιλέξει εμφάνιση ταξινομημένη κατά ονοματεπώνυμο, θα χρειαστεί να κάνετε ταξινόμηση του πίνακα θέσεων κατά ονοματεπώνυμο και μετά την εμφάνιση της λίστας κρατημένων θέσεων να επαναφέρετε τον πίνακα στην προηγούμενη κατάστασή του, με εκ νέου ταξινόμησή του κατά αριθμό θέσης (αξιοποιώντας τον αριθμό θέσης που είναι αποθηκευμένος στο struct κάθε επιβάτη). Χρησιμοποιήστε αλγόριθμο ταξινόμησης της επιλογής σας.



Κατά την έξοδο από το πρόγραμμά σας, φροντίστε να αποθηκεύονται εκ νέου η πινακίδα κυκλοφορίας του λεωφορείου, ο αριθμός θέσεων του λεωφορείου και οι κρατημένες θέσεις του ταξινομημένες κατά ονοματεπώνυμο στο αρχείο bus.txt. Για να το πετύχετε αυτό

- κάνετε ταξινόμηση του πίνακα θέσεων κατά ονοματεπώνυμο,
- εισάγετε, με αναδρομικό αλγόριθμο, κάθε στοιχείο (struct) του ταξινομημένου πίνακα που αφορά σε κρατημένη θέση στο τέλος απλά συνδεδεμένης λίστας για την οποία υπάρχει δείκτης μόνο προς την κεφαλή της (τον πρώτο κόμβο της),
- σαρώστε τη λίστα αυτή από την αρχή προς το τέλος και αποθηκεύστε κάθε κόμβο της στο αρχείο bus.txt.

Π.χ. Έστω η ακόλουθη αρχικά κενή συνδεδεμένη λίστα (όπου Head ο δείκτης προς την κεφαλή της):

Head → NULL

Αν μετά την ταξινόμηση των κρατημένων θέσεων του πίνακα θέσεων κατά ονοματεπώνυμο, η σειρά των επιβατών είναι:

Anestopoulos Dimitris  
Barbagiannis Hlias  
Kyriakou Andreas

οι διαδοχικές διαμορφώσεις της συνδεδεμένης λίστας μετά την εισαγωγή ενός επιβάτη κάθε φορά θα είναι:

Head → Anestopoulos Dimitris → NULL

Head → Anestopoulos Dimitris → Barbagiannis Hlias → NULL

Head → Anestopoulos Dimitris → Barbagiannis Hlias → Kyriakou Andreas → NULL

Σημειώνεται ότι θα μπορούσε να παρακαμφθεί η χρήση της συνδεδεμένης λίστας για την αποθήκευση με τη ζητούμενη σειρά ταξινόμησης των κρατημένων θέσεων στο αρχείο (αφού θα έχει προηγηθεί ταξινόμηση του πίνακα θέσεων κατά ονοματεπώνυμο). Η χρήση τόσο της συνδεδεμένης λίστας όσο και αναδρομικού αλγορίθμου εισαγωγής στο τέλος της συνδεδεμένης λίστας ζητούνται για εκπαιδευτικούς λόγους.

### Θέμα 3: Υλοποίηση Παραλλαγής Παιχνιδιού SET

Κατασκευάστε πρόγραμμα στη γλώσσα προγραμματισμού C, το οποίο θα υλοποιεί μια παραλλαγή του παιχνιδιού SET<sup>1</sup>. Το παιχνίδι είναι ένα επιτραπέζιο παιχνίδι καρτών που παίζεται σε γύρους από έναν ή περισσότερους παίκτες (έστω 3). Για την υλοποίηση του συγκεκριμένου ερωτήματος θεωρήστε ότι τα δεδομένα των χρηστών είναι αποθηκευμένα από τη σχεδίαση του προγράμματος και είναι τα εξής:

- ✓ Όνομα (name) → ΜΕΓΕΘΟΣ 17 ΧΑΡΑΚΤΗΡΕΣ.
- ✓ Βαθμολογία (score) → ΑΚΕΡΑΙΟΣ ΑΡΙΘΜΟΣ ΜΕ ΕΛΑΧΙΣΤΗ ΤΙΜΗ 0 ΚΑΙ ΜΕΓΙΣΤΗ 255. ΑΡΧΙΚΗ ΤΙΜΗ 0
- ✓ Μοναδικό χαρακτήρας αναγνώρισης (id) → Ο ΧΑΡΑΚΤΗΡΑΣ ΑΝΑΓΝΩΡΙΣΗΣ, ΜΟΝΑΔΙΚΟΣ ΓΙΑ ΚΑΘΕ ΠΑΙΧΤΗ

Χρησιμοποιήστε τα παρακάτω στοιχεία:

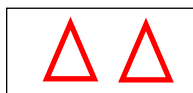
Id παίχτη	a	g	p
Όνομα παίχτη	Marina Andreou	Kostas Grammenos	Maria Perdika

<sup>1</sup> Μία υλοποίηση του παιχνιδιού μπορεί να βρεθεί στην σελίδα <http://www.setgame.com/set/puzzle>



Κάθε κάρτα έχει 4 ιδιότητες (**χρώμα**, **σχήμα**, **αριθμό** και **υφή**), ενώ κάθε ιδιότητα μπορεί να λάβει μία από τρεις διαθέσιμες τιμές, ειδικότερα **χρώμα**: { **r**: κόκκινο, **g**: πράσινο, **b**: μπλε }, **σχήμα**: { **c**: κύκλος, **t**: τρίγωνο, **r**: ρόμβος }, **αριθμός** {1,2,3} και **υφή**: { **b**: πλήρες, **h**: γραμμοσκιασμένο, **e**: κενό-μόνο περίγραμμα }.

Για παράδειγμα μία κάρτα που αναπαρίσταται από την τετράδα [rt2e] αντιστοιχεί στην κάρτα με τις εξής ιδιότητες: [2 κόκκινα τρίγωνα με περίγραμμα]



Σε κάθε γύρο από τις  $3^4=81$  πιθανές διαφορετικές κάρτες εμφανίζονται τυχαία 12 κάρτες σε ένα πλέγμα  $3 \times 4$ . : Δεν απαιτείται η γραφική αναπαράσταση των καρτών στην οθόνη του υπολογιστή πέρα από την προβολή της αναπαράστασης για κάθε κάρτα  $[v1,v2,v3,v4]$  όπου νί η τιμή της κάρτας για την ιδιότητα *i*. Δηλαδή μία πιθανή προβολή του πλέγματος θα μπορούσε να είναι

[r, t, 1, b] [g, r, 2, h] [b, r, 2, e] [r, c, 3, h]  
[g, c, 1, h] [b, t, 3, b] [r, c, 2, h] [b, t, 2, e]  
[g, r, 1, e] [r, c, 3, e] [g, t, 1, b] [b, r, 3, b]

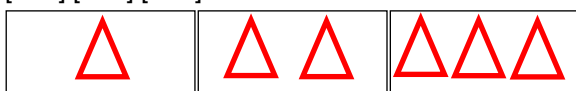
Οι παίκτες παρατηρούν τις κάρτες και προσπαθούν να εντοπίσουν τριάδες καρτών που για κάθε μία από τις 4 ιδιότητες, είτε όλες έχουν την ίδια τιμή, είτε όλες έχουν διαφορετικές τιμές. Θα πρέπει να σημειωθεί ότι δεν μπορούν να υπάρξουν δύο όμοιες κάρτες ταυτόχρονα στο πλέγμα.

Παραδείγματα «ΣΕΤ» αποτελούν οι παρακάτω τριάδες καρτών:

- 1) [rt2e] [gt2e] [bt2e]



- 2) [rt1e] [rt2e] [rt3e]



- 3) [gc1b] [br1h] [rt1e]



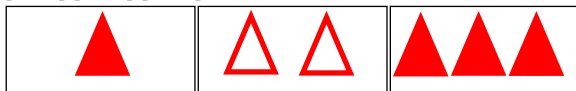
Ενώ παραδείγματα από τριάδες καρτών που δεν αποτελούν «ΣΕΤ» είναι τα παρακάτω:

- 1) [gt2e] [gt1e] [bt2e]



Η πρώτη κάρτα έχει την ιδιότητα αριθμός κοινή με την τρίτη και διαφορετική με την δεύτερη. Για να ήταν σετ έπρεπε είτε όλες οι κάρτες να έχουν τον ίδιο αριθμό σχημάτων είτε όλες διαφορετικό

- 2) [rt1b] [rt2e] [rt3b]



Αντίστοιχο πρόβλημα με την υφή των σχημάτων

3) [gc1b] [br1h] [gc1e] ]



Οι κάρτες έχουν όλες διαφορετική υφή και ίδιο αριθμό σχημάτων. Το πρόβλημα παρατηρείται στο είδος των σχημάτων και στο χρώμα.

**ΖΗΤΟΥΜΕΝΑ:** Υλοποιήστε πρόγραμμα κατά την εκτέλεση του οποίου:

- Να επιλέγονται 12 τυχαίες κάρτες από τις 81 πιθανές και να εμφανίζονται στην οθόνη χρησιμοποιώντας ένα πλέγμα 3 γραμμών και 4 στηλών. Να υπάρξει μέριμνα ώστε να μην εμφανίζεται δύο φορές η ίδια κάρτα.
- Ο έλεγχος του προγράμματος να περιμένει είτε τον χαρακτήρα αναγνώρισης του παίχτη είτε τον αριθμό 0 που σημαίνει και τον τερματισμό εκτέλεσης του προγράμματος. Θεωρήστε ότι το id ενός παίχτη καθορίζεται πλήρως από έναν χαρακτήρα. Ο παίχτης που επιλέγεται θεωρείται ενεργός. Με κάθε άλλη είσοδο οι κάρτες να ανακατεύονται και να εμφανίζεται μία νέα τυχαία 12αδα καρτών.
- Να υλοποιηθεί συνάρτηση που με δεδομένο τις θέσεις τριών καρτών που εμφανίζονται στο πλέγμα να αποφαινεται εάν το σύνολο των καρτών αποτελεί σερ ή όχι.
  - Σημ: για να ορίσουμε την θέση μίας κάρτας στο πλέγμα θα πρέπει να επιλέξουμε τις συντεταγμένες x,y της κάρτας στο πλέγμα ξεκινώντας από το 0. Πχ αν επιθυμούσαμε να επιλέξουμε την κάρτα [r, c, 2, h] που βρίσκεται στην 2<sup>η</sup> γραμμή και στην 3<sup>η</sup> στήλη του πλέγματος θα επιλέγαμε x=1, y=2  
Θεωρήστε ότι ο παίχτης θα δίνει είσοδο αριθμό. Χειριστείτε κατάλληλα την περίπτωση που ο αριθμός είναι εκτός επιτρεπτών ορίων.
- Αφού επιλεγεί παίχτης ως ενεργός να του ζητείται να επιλέξει 3 κάρτες. Εάν το σύνολο των καρτών που επέλεξε ο παίχτης αποτελεί ΣΕΤ να αυξάνεται η βαθμολογία του κατά δύο μονάδες και να επιλέγονται 12 νέες κάρτες. Αν δεν αποτελεί ΣΕΤ να μειώνεται η βαθμολογία του κατά μία μονάδα με ελάχιστο όριο το μηδέν και να προβάλλεται η ίδια 12αδα καρτών.
- Η διαδικασία ολοκληρώνεται όταν αντί κωδικού παίχτη εισάγεται η επιλογή μηδέν ή όταν κάποιος παίχτης καταφέρει να συγκεντρώσει 10 βαθμούς. Πριν τον τερματισμό προβάλλονται οι πληροφορίες όλων των παιχτών, ενώ στην περίπτωση που κάποιος παίχτης έχει συγκεντρώσει 10 πόντους θα τυπώνεται στο τέλος το μήνυμα :  
“ο nikhts einai o [name]”,  
όπου [name] το όνομα του παίχτη.

Σημ 1<sup>η</sup> : Για την τυχαία επιλογή<sup>2</sup> των 12 καρτών από τις 81 πιθανές υλοποιήστε κατάλληλη συνάρτηση με όνομα getNextCard(...) και κατάλληλα ορίσματα της επιλογής σας, χρησιμοποιώντας την συνάρτηση rand(). Υπενθυμίζεται ότι οι κάρτες που επιλέγονται θα πρέπει να είναι μοναδικές. Μην ξεχνάτε την κλήση της συνάρτησης srand(time(NULL)); στην main για την ορθή αρχικοποίηση της συνάρτησης rand και την συμπερίληψη του αρχείου κεφαλίδας βιβλιοθήκης <time.h> στην αρχή του προγράμματος.

<sup>2</sup> [https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_rand.htm](https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm)  
<http://stackoverflow.com/questions/822323/how-to-generate-a-random-number-in-c>  
<http://stackoverflow.com/questions/10776073/random-double-between-min-and-max>



### Γνωστικά Αντικείμενα ανά θέμα

	Θ1	Θ2	Θ3
Πέρασμα παραμέτρων με αναφορά	X	X	X
Δείκτες		X	X
Συμβολοσειρές	X	X	
Αναδρομή		X	
Πολυδιάστατοι πίνακες			X
Δυναμικοί πίνακες		X	
Σύνθετες δομές δεδομένων	X	X	X
Απλά συνδεδεμένες λίστες - ουρές		X	
Αναζήτηση	X	X	
Ταξινόμηση		X	
Ψευδοτυχαίοι αριθμοί			X

ΚΡΙΤΗΡΙΑ ΑΞΙΟΛΟΓΗΣΗΣ	
Θέμα 1: Νέο σύστημα καταγραφής κρατημένων θέσεων	25
Θέμα 2: Επέκταση συστήματος καταγραφής κρατημένων θέσεων	30
Θέμα 3: Υλοποίηση Παραλλαγής Παιχνιδιού	45
Εικόνα εργασίας - σχολιασμός	10
<b>ΣΥΝΟΛΟ</b>	<b>110</b>
Ο συνολικός βαθμός θα διαιρεθεί δια 10, ώστε να προκύψει ο τελικός βαθμός της εργασίας. Μέγιστος βαθμός το 10.	

**Καλή Επιτυχία!!!**